

WHAT IS CLAIMED IS:

Sub
A3
1. A system for enabling code execution from non executable memory, comprising:

- i. An executing entity, for executing code for a host system;
- ii. A non-executable memory component, for storing system code and data; and.
- iii. An executable memory component, for operating as a memory buffer for executing said code, such that a portion of contents of said non-executable memory component is located within said executable memory component, and said portion of contents of said non-executable memory component emulates executable functions of said executable memory component.

2. The system of claim 1, wherein said executable memory component employs a downloading mechanism for downloading requested data from said non-executable memory component to said executable memory component, such that data addresses requested by said executing entity are downloaded to said executable memory component.

3. The system of claim 1, wherein said non-executable memory component is selected from the group consisting of NAND flash components, serial EEPROM and flash memory components, such that said non-executable memory component functions as an executable memory.

Sub
A4
4. A system for executing code using non-executable memory, comprising:

- i. An executing entity, for executing code;
- ii. A non-executable memory component, for storing said code and data; and.
- iii. A plurality of executable memory components that operate as multiple memory buffers for preventing memory lockage for accesses to said data during download operations of said code.

5. The system of claim 4, wherein each of said plurality of memory buffers include download logic and memory buffer space.

- Sub*
- 6. A method for executing code using non-executable memory, comprising the steps of:
 - i. providing executable memory, for buffering at least one code request from an executing entity;
 - ii. providing a non-executable memory, for storing executable code;
 - iii. downloading at least a portion of said executable code to said executable memory, for emulating executable functions of said executable memory;
 - iv. executing at least one said code request from said executable memory; and
 - v. buffering an execution of contents of said non-executable memory in said executable memory.

7. The method of claim 6, further comprising the steps of:

- a) managing at least one set of instructions to guarantee availability of said contents in an executable buffer; and
- b) supplying a busy signal in cases where said contents are not available, such that the executable entity delays the read cycle until said contents are available.

8. The method of claim 6, wherein said downloading at least a portion of said executable code includes the steps of:

- (a) querying said executable memory for data; and
- (b) when queried address of said data is only available in non-executable memory, initiating a download operation from a required location of said non-executable memory, to a buffer area of said executable memory.

9. The method of claim 6, such that step iv further includes:

- I. providing a plurality of executable memory buffers for preventing said portion of said non-executable memory from being locked for accesses during *said* downloading operation;
- II. loading said executable code to one of said plurality of executable memory buffers; and
- III. maintaining at least one of additional said executable memory buffers, to be accessible to said host system and executable by said *host system*.

add 47
21

add
D1

2025-03-06